Ruby on Rails para desarrollo web

Edwin Caldon
Codescrum S.A.S.

Agenda

1. Ruby
2. Rails

Ruby

Lenguaje de programación Orientado a Objetos, de propósito general, de alto nivel e **interpretado** (Similar a Python)

Yukihiro "Matz" Matsumoto (1995)

Prueba **Ruby** en el navegador web:
https://ruby.github.io/TryRuby/

ruby -v
Versión actual: 2.6

Ruby

Consola interactiva de Ruby

Interactive Ruby Shell (irb)

```
irb(main):001:0> puts “ProDev”
irb(main):001:0> 2 + 4
irb(main):001:0> number = 5
irb(main):001:0> puts “number: #{number}”
irb(main):001:0> colors = { "red" => 'rojo', "green" => 'verde'}
irb(main):078:0> colors.each do |clave, valor|
irb(main):079:1* print clave, " es ", valor, "\n"
irb(main):080:1> end
```

Ruby

Consola interactiva de Ruby

Interactive Ruby Shell (irb)

```
irb(main):001:0> 11234.methods
irb(main):001:0> 11234.digits
irb(main):001:0> 11234.negative?
irb(main):001:0>
irb(main):001:0> 11234.class
irb(main):001:0> 1234.class.ancestors
```

Ruby

Interactive Ruby Shell (irb)

```
irb(main):011:0> var = :symbol
=> :symbol
irb(main):012:0> var2 = :symbol
=> :symbol
irb(main):013:0> var.object_id
=> 879068
irb(main):014:0> var2.object_id
=> 879068
irb(main):014:0> colors = { :red => 'rojo', :green => 'verde'}
```

Ruby: Iteradores

Consola interactiva de Ruby

Interactive Ruby Shell (irb)

```
irb(main):001:0> a = %w{ ruby python c c++ }
irb(main):001:0> a.each { |lenguaje| puts lenguaje }
```

```
irb(main):015:0> for l in a do
irb(main):016:1* puts l
irb(main):017:1> end
```

```
irb(main):001:0> 5.times {  print "*" }
irb(main):001:0> 3.upto(6) {|i|  print i }
```

Ruby: Iteradores

irb(main):001:0> [1,2,3]**.select**(&:**even?**)

irb(main):001:0> 2.methods
irb(main):001:0> 2.**even?**

Qué es **:even**

```
result = []
[1,2,3,4].each do |i|
  if i % 2 == 0
    result << i
  end
end
print result
```

irb(main):001:0> [1,2,3].select{|num| num.even?}

Ruby: Iteradores

Enumerador:

```
irb(main):001:0> 1.upto(10) do |i| puts i end
irb(main):001:0> 10.downto(1) do puts i end
```

Array:

```
irb(main):080:1> [*1..10].each{|i| puts "i=#{i}" }
```

Range:

```
irb(main):001:0> (1..10).each{|i| puts "i=#{i}" }
irb(main):001:0> (1..10).to_a
```

Ruby: Iteradores

Más en:
https://ruby-doc.org/core-2.6.3/Enumerable.html

```
irb(main):001:0> [1,2,3].any? {|num| num.even?}
=> true

irb(main):001:0> [1,2,3].all? {|num| num.even?}
=> false


irb(main):001:0> defined? [1,2,3].all?
```

Ruby: Funciones

```
irb(main):001:0> def double(val)
irb(main):001:0>   val * 2
irb(main):001:0> end
=> :double
irb(main):001:0> double(20)
=> 40
```

Ruby: Funciones

Argumento por defecto y retorno por defecto

En cualquier editor, crear el archivo:
01.functions.rb

```ruby
def double(val=10)
  val * 2
end

puts double(20)
puts double
```

$ ruby 01.functions.rb

```
codescrum@codescrum:~/ruby_apps$ ruby 01.functions.rb
40
20
```

Ruby: Funciones

- Usando la función **double(val)** como bloque explicito:

```
irb(main):001:0> [1,2,3].each(&:double)
```

Proc, bloques y arrays (tema para otra sesión)

```
irb(main):002:0> [1,2,3].each{ |n| double(n) }
```

```
irb(main):003:0> [1,2,3].map{ |n| double(n) }
```

Ruby: Funciones

```ruby
def change_case(x)
  x.upcase
end

def change_case!(x)
  x.upcase!
end

a = 'text text text'
puts change_case(a)
puts a
puts change_case!(a)
puts a
```

```
codescrum@codescrum:~/ruby_apps$ ruby 01.functions.rb
TEXT TEXT TEXT
text text text
TEXT TEXT TEXT
TEXT TEXT TEXT
```

Ruby: Clases

```ruby
class Article
  def to_s
    "Article "
  end

  private
  def change_name(name)
    puts "New name is #{name}"
  end
end

article = Article.new
puts article.to_s
article.change_name("Codescrum")
```

Definición de la clase Article

Ruby: Clases

```
codescrum@codescrum:~/ruby_apps$ ruby 02.classes.rb
Article
Traceback (most recent call last):
02.classes.rb:15:in `<main>': private method `change_name
codescrum@codescrum:~/ruby_apps$
```

Error el llamar al método privado

Ruby: Clases

Constructor

def initialize

- Variables de Instancia: @name

```ruby
class Article
  def initialize(name, author)
    @name = name
    @author = author
  end

  def to_s
    "Article: #{@name} by #{@author}"
  end

  private
  def change_name(name)
    @name = name
    puts "New name is #{name}"
  end
end

article = Article.new("Ruby on Rails", "Edwin Caldon")
puts article.to_s
```

codeScrum

Ruby: Clases

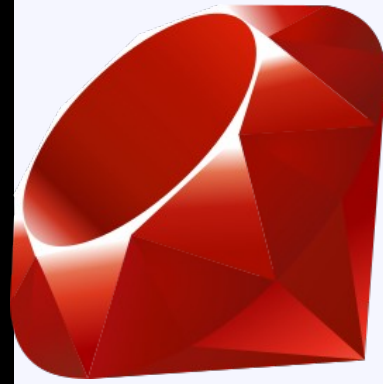
Metodos getter

```
def name
  @name
end
```

y setter

```
def name=(name)
  @name = name
end
```

```ruby
class Article
  def initialize(name, author)
    @name = name
    @author = author
  end

  def to_s
    "Article: #{@name} by #{@author}"
  end

  def name
    @name
  end

  def name=(name)
    change_name(name)
  end

  private
  def change_name(name)
    @name = name
  end
end

article = Article.new("Ruby on Rails", "Edwin Caldon")
puts article.to_s
puts article.name
article.name = "Gemas RoR"
puts article.to_s
```



Ruby: Clases

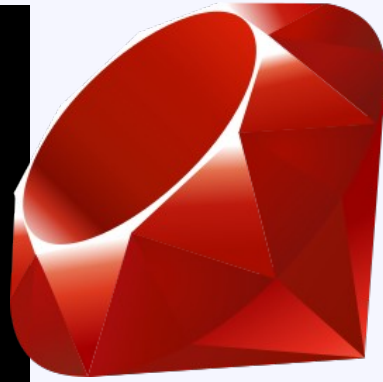
Accessors:

attr_accessor
- getter
- setter

attr_reader
- getter

attr_writer
- setter

```ruby
class Article
  attr_accessor :author, :name

  def initialize(name, author)
    @name = name
    @author = author
  end

  def to_s
    "Article: #{@name} by #{@author}"
  end

  private
  def change_name(name)
    @name = name
  end
end

article = Article.new("Ruby on Rails", "Edwin Caldon")
puts article.to_s
puts article.author
article.author = "Cristian O."
puts article.to_s
```



Ruby: Clases

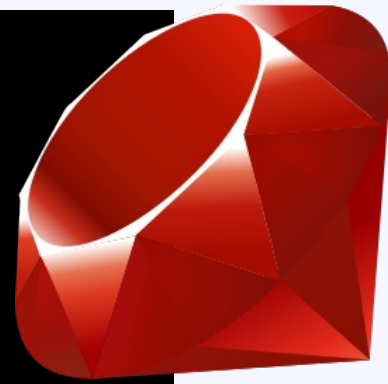
Variables de instancia:
- @name

Variables de clase:
- @@instances

Métodos de instancia:
- def to_s

Métodos de clase:
- def self.instances

```ruby
class Article
  attr_accessor :author, :name

  @@instances = 0
  def initialize(name, author)
    @name = name
    @author = author
    @@instances += 1
  end

  def to_s
    "Article: #{@name} by #{@author}"
  end

  def self.instances
    @@instances
  end
end

article = Article.new("Ruby on Rails", "Edwin Caldon")
puts article
puts Article.instances

new_article = Article.new("Docker", "Cristian O.")
puts new_article
puts Article.instances
```



codeScrum

Ruby: Gemas en ruby

Gem: librería que empaqueta código ruby:

- Formato común para compartir librerías y herramientas de Ruby
- Dependencia que serán instaladas automáticamente con la gema.

https://rubygems.org/

gem install <nombre gema>

Ruby: Gemas en ruby

Bundle (https://bundler.io/): herramienta para administrar las dependencias de tu proyecto ruby.

gem administra las dependencias de la gema en sí; pero tu proyecto no es una gema, así nace bundle:

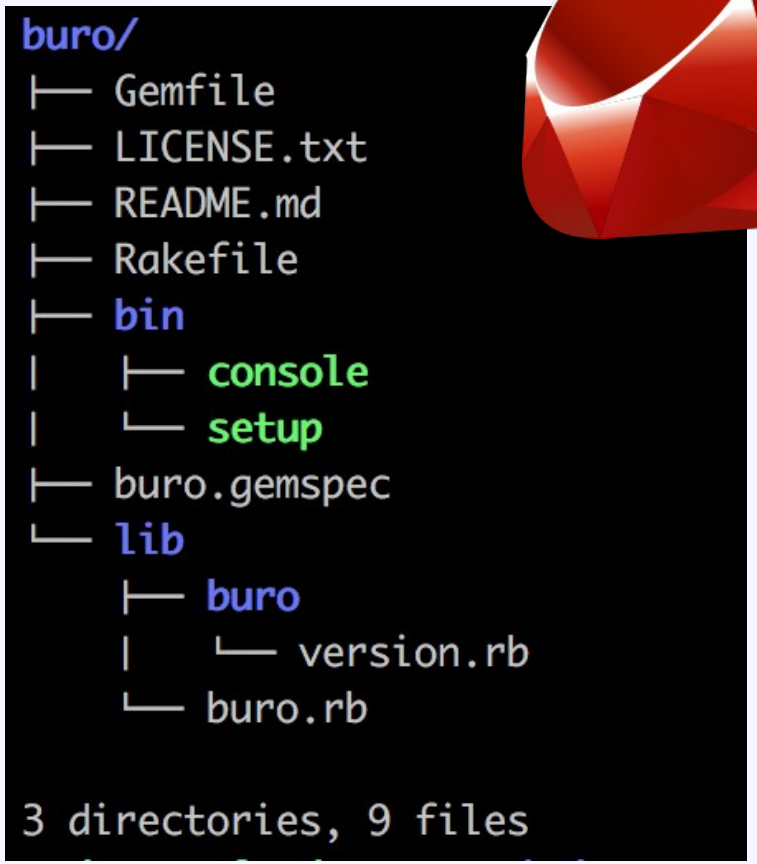
- gem install bundler
- bundle -v
  - 2.0.2 o superior
- bundle install
- bundle upgrade

https://bundler.io/docs.html

Ruby: Gemas en ruby

Vamos a crear una gema:

- **bundle gem buro**
  - Gemfile (dependencias)
  - LICENSE.txt
  - buro.gemspec (configuraciones)
  - lib/buro (código de la gema)
    - lib/buro/version.rb
  - lib/buro.rb (archivo principal)



Ruby on Rails (RoR)

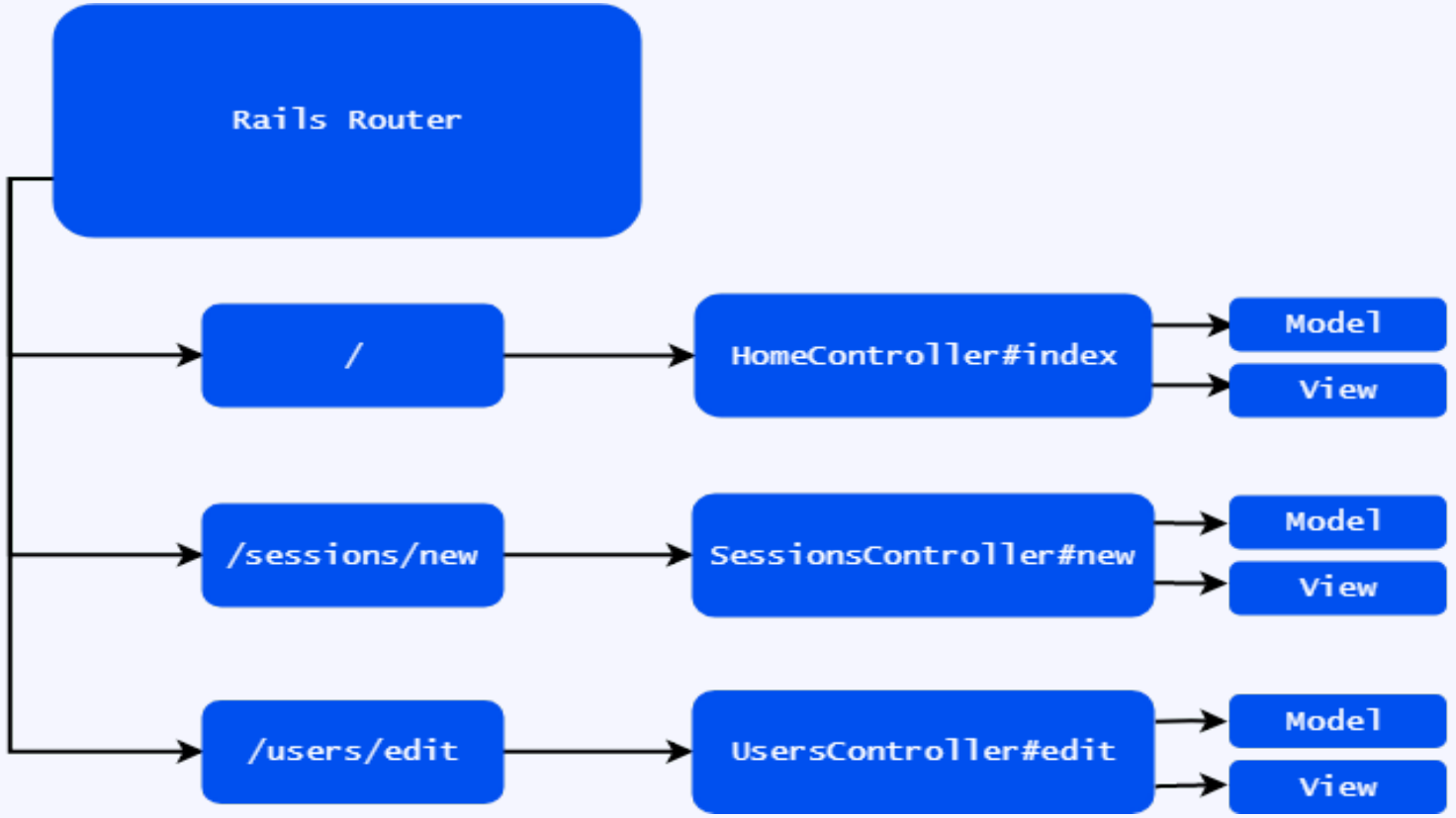
Ruby on Rails (RoR) es un framework web basado en Ruby (es como un Django sobre Python)

David Heinemeier Hansson (DHH) (2005)

Quiénes usan RoR:
Github, Shopify, Ask.fm, Kickstarter, Scribd, ConvertKit, Twitch, Instacart, Zendesk, SoundCloud

Ruby on Rails (RoR)

Patrón MVC:



codeScrum

Ruby on Rails (RoR)

Filosofía de Rails:

- **Don't Repeat Yourself**: **DRY**

  No escribir la información una y otra vez.

- **Convention Over Configuration**:

  Si es algo que se usa muchas veces, Rails ya lo tiene implementado por usted.

RoR: instalación

Linux:

1) Instalar y configurar **rbenv**:

```
codescrum@codescrum:~$ cd ~/.rbenv && src/configure && make -C src
make: Entering directory '/home/codescrum/.rbenv/src'
make: '../libexec/rbenv-realpath.dylib' is up to date.
make: Leaving directory '/home/codescrum/.rbenv/src'
codescrum@codescrum:~/.rbenv$ cd
codescrum@codescrum:~$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bash_profile
codescrum@codescrum:~$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
```

```
codescrum@codescrum:~$ rbenv install -l
Available versions:
  1.8.5-p52
  1.8.5-p113
  1.8.5-p114
  1.8.5-p115
  1.8.5-p231
```

RoR: instalación

Linux:

1) Instalar y configurar **rbenv**: https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-with-rbenv-on-ubuntu-18-04
    - **rbenv** es un administrador de versiones de ruby (como virtualenv o pyenv de python)

```
codescrum@codescrum:~$ gem env home
/home/codescrum/.rbenv/versions/2.6.3/lib/ruby/gems/2.6.0
```

```
codescrum@codescrum:~$ rbenv root
/home/codescrum/.rbenv
```

Asegurarse de tener instalado las versiones correctas de ruby:

rbenv install **2.6.3**
rbenv install -l

RoR: instalación

Linux:

1) Instalar y configurar **rbenv**:
   https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-with-rbenv-on-ubuntu-18-04

rbenv plugins

- https://github.com/rkh/rbenv-update
- https://github.com/rbenv/rbenv-vars

Gemas en Ruby on Rails (RoR)

- > rails new [nombre-proyecto] -T

```
codescrum@codescrum:~/rails_apps/test_rails$ ll
total 88
drwxrwxr-x 13 codescrum codescrum 4096 Jul 23 16:04 ./
drwxrwxr-x  3 codescrum codescrum 4096 Jul 23 15:59 ../
drwxrwxr-x  7 codescrum codescrum 4096 Jul 23 15:59 .git/
-rw-rw-r--  1 codescrum codescrum  665 Jul 23 15:59 .gitignore
-rw-rw-r--  1 codescrum codescrum    6 Jul 23 15:59 .ruby-version
-rw-rw-r--  1 codescrum codescrum 1732 Jul 23 16:04 Gemfile
-rw-rw-r--  1 codescrum codescrum 4915 Jul 23 16:04 Gemfile.lock
-rw-rw-r--  1 codescrum codescrum  374 Jul 23 15:59 README.md
-rw-rw-r--  1 codescrum codescrum  227 Jul 23 15:59 Rakefile
drwxrwxr-x 11 codescrum codescrum 4096 Jul 23 15:59 app/
drwxr-xr-x  2 codescrum codescrum 4096 Jul 23 15:59 bin/
drwxrwxr-x  5 codescrum codescrum 4096 Jul 23 15:59 config/
-rw-rw-r--  1 codescrum codescrum  130 Jul 23 15:59 config.ru
drwxrwxr-x  3 codescrum codescrum 4096 Jul 23 16:49 db/
drwxrwxr-x  4 codescrum codescrum 4096 Jul 23 15:59 lib/
drwxrwxr-x  2 codescrum codescrum 4096 Jul 23 16:00 log/
-rw-rw-r--  1 codescrum codescrum  240 Jul 23 15:59 package.json
drwxrwxr-x  2 codescrum codescrum 4096 Jul 23 15:59 public/
drwxrwxr-x  2 codescrum codescrum 4096 Jul 23 15:59 storage/
drwxrwxr-x  6 codescrum codescrum 4096 Jul 23 16:04 tmp/
drwxrwxr-x  2 codescrum codescrum 4096 Jul 23 15:59 vendor/
```

Gemas en Ruby on Rails (RoR)

- Archivo **Gemfile**, dependencias del proyecto:
  - Ruby ‘2.6.3’

‘Rails’, ‘6.0.0.rc1’ (exacta)

‘Rails’, ‘>= 6.0.0.rc1’ (igual o mayor)

‘Rails’, ‘>= 6.0.0.rc1’, ‘< 6.0.0.rc2’
(igual o mayor, pero menor que)

‘Rails’, ‘~= 6.0.0.rc1’
(igual o mayor, pero menor que 6.0.0.rc2)

```
source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby '2.6.3'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '~> 6.0.0.rc1'
# Use sqlite3 as the database for Active Record
gem 'sqlite3', '~> 1.4'
# Use Puma as the app server
gem 'puma', '~> 3.11'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5'
# Transpile app-like JavaScript. Read more: https://github.com/rc
# gem 'webpacker', '~> 4.0'
# Turbolinks makes navigating your web application faster. Read m
gem 'turbolinks', '~> 5'
# Build JSON APIs with ease. Read more: https://github.com/rails/
gem 'jbuilder', '~> 2.5'
# Use Redis adapter to run Action Cable in production
# gem 'redis', '~> 4.0'
# Use Active Model has_secure_password
# gem 'bcrypt', '~> 3.1.7'
```

Gemas en Ruby on Rails (RoR)

Si se omite la versión en una gema, se instala la última disponible:

gem ‘rails’ (6.0.0.rc2)

https://github.com/rails/rails

Versionamiento semántico:
https://semver.org/lang/es/

Se puede especificar otra fuente:
gem 'devise', git: "https://github.com/plataformatec/devise"

Gemas en Ruby on Rails (RoR)

En el archivo **Gemfile** se pueden agrupar las dependencias por ambiente/grupo de desarrollo:

- :development
- :test
- :production
- ~~:assets~~
- :staging

Gemfile es administrado por bundle:

- bundle install

```
group :development, :test do
  gem 'rb-readline'
  gem 'spirit_hands'
end

group :development do
  # Access an interactive console on excepti
  gem 'web-console', '>= 3.3.0'
  gem 'listen', '>= 3.0.5', '< 3.2'
  # Spring speeds up development by keeping
  gem 'spring'
  gem 'spring-watcher-listen', '~> 2.0.0'
  # syncing dbs to and from heroku
  gem 'parity'
end

group :test do
  # Adds support for Capybara system testing
  gem 'capybara', '>= 2.15'
  gem 'selenium-webdriver'
  gem 'webdrivers'
  gem 'rspec-rails'
  # # Easy installation and use of chromedri
  # gem 'chromedriver-helper'
end
```

Gemas en Ruby on Rails (RoR)

Grupo o environment, en una **rails console**:

```
irb(main):001:0> Rails.env.production?
irb(main):001:0> Rails.env.development?
irb(main):001:0> Rails.env.test?
irb(main):001:0> Rails.env
irb(main):001:0> ENV
irb(main):001:0>
```

El grupo/ambiente se define en una variable de sistema:

```
RAILS_ENV=development bundle exec rake db:drop
RAILS_ENV=development bundle exec rake db:create
RAILS_ENV=development bundle exec rake db:migrate
RAILS_ENV=development bundle exec rake db:populate
```

Gemas en Ruby on Rails (RoR)

Gemas básicas para un proyecto rails:

- gem 'rails', '~> 6.0.0.rc2'
- gem 'pg', '>= 0.18', '< 2.0'
  - gem 'sqlite3', '~> 1.4'
- gem 'puma', '~> 3.11'
- gem 'sass-rails', '~> 5'
- gem 'devise', git: "https://github.com/plataformatec/devise"
- gem 'haml'
- ***Development:***
  - gem 'jazz_fingers'
  - gem 'spirit_hands'
  - gem 'spring'
  - gem 'listen'

Gemas en Ruby on Rails (RoR)

Gemas básicas en un proyecto rails:

- ***Test:***
  - gem 'capybara', '>= 2.15'
  - gem 'selenium-webdriver'
  - gem 'rspec-rails'

Gemas en Ruby on Rails (RoR)

- pg (https://rubygems.org/gems/pg): es la interfaz Ruby para el GBD PostgreSQL

```ruby
#!/usr/bin/env ruby

require 'pg'

# Output a table of current connections to the DB
conn = PG.connect( dbname: 'codescrum', host: 'localhost', user: 'codescrum', password: 'codescrum' )
conn.exec( "SELECT * FROM code" ) do |result|
  puts "     ID | Username            | Last_login"
  result.each do |row|
    puts " %7d | %-16s | %s " %
      row.values_at('id', 'username', 'last_login')
  end
end
```

Gemas en Ruby on Rails (RoR)

```
codescrum@codescrum:~/rails_apps$ ruby connect_postgresDB.rb
 ID | Username                | Last_login
  3 | edwin.caldon            | 2019-08-09 20:05:01.84056
```

En un proyecto rails, las conecciones a postgres se hacen en:

config/database.yml

Gemas en Ruby on Rails (RoR)

- sqlite3 (https://rubygems.org/gems/sqlite3/) : interfaz ruby con el motor de BD SQLite3:
    - https://www.rubydoc.info/gems/sqlite3/1.4.1

```
codescrum@codescrum:~/rails_apps$ ruby connect_sqlite3DB.rb
========== numbers table ===========
["one", 1]
["two", 2]
============ students table ================
["Jane", "me@janedoe.com", "A", "http://blog.janedoe.com"]
```

Gemas en Ruby on Rails (RoR)

- puma (https://rubygems.org/gems/puma/) servidor web para aplicaciones ruby:
    - https://www.rubydoc.info/gems/puma

Cuando se ejecuta ***rails server*** se lanza ***puma***.

En un proyecto rails las configuraciones de puma se hacen en:
config/puma.rb

Gemas en Ruby on Rails (RoR)

- sass-rails (https://github.com/rails/sass-rails) motor de renderizado css para RoR:

sass - old

```
$color: red

=my-border($color)
  border: 1px solid $color


body
  background: $color
  +my-border(green)
```

scss - new

```
$color: red;

@mixin my-border($color) {
  border: 1px solid $color;
}

body {
  background: $color;
  @include my-border(green);
}
```

Gemas en Ruby on Rails (RoR)

Etiquetas anidadas:

```
#error_explanation {
  width: 450px;
  border: 2px solid red;
  h2 {
    text-align: left;
  }
  ul li {
    font-size: 12px;
    list-style: square;
  }
}
```

Gemas en Ruby on Rails (RoR)

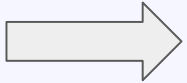
- Devise (https://github.com/plataformatec/devise) autenticación para proyectos Rails:
  - OmniAuth (https://github.com/omniauth/omniauth/wiki)
    - omniauth-github
    - omniauth-google-oauth2
    - https://github.com/omniauth/omniauth/wiki/List-of-Strategies

Gemas en Ruby on Rails (RoR)

- haml (http://haml.info/): HTML abstraction markup language:

.html

```
<section class="container">
  <h1><%= post.title %></h1>
  <h2><%= post.subtitle %></h2>
  <div class="content">
    <%= post.content %>
  </div>
</section>
```



.haml

```
%section.container
  %h1= post.title
  %h2= post.subtitle
  .content
    = post.content
```

Gemas en Ruby on Rails (RoR)

- 'jazz_fingers' (https://github.com/plribeiro3000/jazz_fingers):
  - Pry (http://pryrepl.org/): es un REPL (read–eval–print loop) para rails.
- 'spirit_hands' (https://rubygems.org/gems/spirit_hands/)
- 'spring' (https://github.com/rails/spring): aplicación rails en segundo plano y se actualiza con cualquier cambio en el código.
- 'Capybara' (https://github.com/teamcapybara/capybara) pruebas funcionales, pruebas de usuario.
- gem 'selenium-webdriver'
- gem 'rspec-rails'

Gema Devise en RoR

- > rails new **findpet**
- Editar Gemfile y agregar la gema
- > bundle install
- > rails g devise:install
- > rails g devise User
- > rails db:migrate
- > rails generate devise:views

- > rake routes

Gema Devise en RoR

- > rail server -b 0.0.0.0 -p 9090
  - rails webpacker:install
  - https://yarnpkg.com/lang/en/docs/install/#debian-stable
- > rails console
  - irb(main):002:0> u = User.**find_by_email**('edycop@gmail.com')
  - irb(main):002:0> u

Más de *rails console* en https://guides.rubyonrails.org/command_line.html#rails-console

Gema Devise en RoR

- Editar *config/routes.rb* y definir una nueva para la página raíz:
  - root '**home#show**' (control#acción)
- > rails g controller home
- Agregar la acción ***show*** al controlador *app/controllers/home_controller.rb*
- Crear la vista en app/views/home/***show***.html.erb

```
<p>
  <% if user_signed_in? %>
    <string>Usuario registrado</strong>
  <% else %>
    <strong>Usuario no registrado</strong>
  <% end %>
</p
```

Gema Devise en RoR

- Complemento de la vista en ***app/views/home/show.html.erb*** :

```
<p>
  <% if user_signed_in? %>
    <string>Usuario registrado</strong>
    Logueado como <strong><%= current_user.email %></strong>.
    <%= link_to "Logout", destroy_user_session_path, method: :delete %>
  <% else %>
    <strong>Usuario no registrado</strong>
    <%= link_to "Sign up", new_user_registration_path  %> |
    <%= link_to "Login", new_user_session_path  %>
  <% end %>
</p>
```

RoR: opciones al crear proyecto rails

> rails new [nombre-proyecto] -T

> rails new [nombre-proyecto] -d postgresql

> rails g model user name:string username:string email:string password_digest:string

> rails g scaffold Post title:string body:text

> rake db:setup

> rake db:create

> rake db:migrate

> rake db:seed

RoR: opciones al crear proyecto rails

> rails server

> rails server -b 0.0.0.0 -p 8090

> rails g controller home

>

Gracias!

Comandos útiles (Postgres)

- Para conectarse a **PostgreSQL** y crear usuario y BD:
  - Editar /etc/postgresql/10/main/postgresql.conf
    - listen_addresses = '*'
  - Editar /etc/postgresql/10/main/pg_hba.conf
    - host all all 0.0.0.0/0 md5
  - sudo postgres
  - psql
    - CREATE ROLE *codescrum* NOINHERIT LOGIN PASSWORD 'codescrum';
    - CREATE DATABASE **codescrum** OWNER *codescrum*;
    - GRANT ALL PRIVILEGES ON DATABASE **codescrum** TO *codescrum*;

Comandos útiles (Postgres)

- En el shell de postgreSQL:
  - Como usuario codescrum del sistema:
    - psql
  - \c codescrum
  - CREATE TABLE code (id serial PRIMARY KEY, username VARCHAR(50) UNIQUE NOT NULL, last_login TIMESTAMP);
  - \d
  - INSERT INTO code(username, last_login) VALUES('edwin.caldon', now());
  - SELECT * FROM code;

  - ALTER TABLE code OWNER TO codescrum; (cambiar propietario de una tabla y debe hacerse como usuario admin [postgres])